(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3676411号
(P3676411)
(45)発行日　平成17年7月27日(2005.7.27)　(24)登録日　平成17年5月13日(2005.5.13)

(51)Int.Cl.[7]　F I
G06F 9/42　G06F 9/42 330R
G06F 9/34　G06F 9/34 330

請求項の数 8　(全 17 頁)

(21)出願番号　特願平7-25821
(22)出願日　平成7年1月23日(1995.1.23)
(65)公開番号　特開平8-44565
(43)公開日　平成8年2月16日(1996.2.16)
審査請求日　平成14年1月21日(2002.1.21)
(31)優先権主張番号　184044
(32)優先日　平成6年1月21日(1994.1.21)
(33)優先権主張国　米国(US)

(73)特許権者　595034134
サン・マイクロシステムズ・インコーポレイテッド
Ｓｕｎ　Ｍｉｃｒｏｓｙｓｔｅｍｓ，　Ｉｎｃ．
アメリカ合衆国　カリフォルニア　９５０５４，　サンタ　クララ，　ネットワーク　サークル　４１５０
(74)代理人　100064621
弁理士　山川　政樹
(72)発明者　ロバート・ヤング
アメリカ合衆国　９４５５５　カリフォルニア州・フレモント・コマース　ドライブ・５７９５

最終頁に続く

(54)【発明の名称】レジスタファイル装置及びレジスタファイルアクセス方法

(57)【特許請求の範囲】
【請求項１】
それぞれ複数のレジスタを備えた複数の論理ウィンドウを備え、少なくとも１つの論理ウィンドウの一部のレジスタが他の論理ウィンドウと共用されているレジスタファイル装置であって、
前記各論理ウィンドウのそれぞれのレジスタを指定するためにそれぞれのレジスタに仮想レジスタ番号が付されているレジスタファイル装置において、
前記複数の論理ウィンドウの中から選択し、それにより、任意の時点でのアクセスを第１の選択された論理ウィンドウに制限する第１のウィンドウ選択手段と、
前記第１の選択された論理ウィンドウのレジスタ群から、所望のレジスタをその仮想レジスタ番号により直接選択する第１のレジスタ選択手段と
を設け、仮想レジスタ番号から、実際のレジスタを示す物理レジスタ番号への変換を行うことなく、レジスタ・ファイルへのアクセスを可能にしたことを特徴とするレジスタファイル装置。
【請求項２】
前記第１の選択された論理ウィンドウは書込みによりアクセスされ、　さらに、前記装置が、
前記複数の論理ウィンドウの中から選択し、それにより、任意の時点でのアクセスを第２の選択された論理ウィンドウに制限する第２のウィンドウ選択手段と、
前記第２の選択された論理ウィンドウのレジスタ群から、所望のレジスタをその仮想レ

ジスタ番号により直接選択する第２のレジスタ選択手段と
を具備しており、この第２の選択された論理ウィンドウが読取りによりアクセスされることを特徴とする請求項１記載の装置。
【請求項３】
任意の時点でアクセス可能である少なくとも１つの大域レジスタをさらに具備する請求項１記載の装置。
【請求項４】
それぞれ複数のレジスタを備えた複数の論理ウィンドウを備え、少なくとも１つの論理ウィンドウの一部のレジスタが他の論理ウィンドウと共用されているレジスタファイル装置であって、さらに
前記各論理ウィンドウのそれぞれのレジスタを指定するためにそれぞれのレジスタに仮想レジスタ番号が付されているレジスタファイル装置のレジスタファイルアクセス方法において、
前記複数の論理ウィンドウの中から選択し、任意の時点でのアクセスを第１の選択された論理ウィンドウに制限する過程と、
前記第１の選択された論理ウィンドウのレジスタ群から、所望のレジスタをその仮想レジスタ番号により直接選択する過程と
から成り、仮想レジスタ番号から、実際のレジスタを示す物理レジスタ番号への変換を行うことなく、レジスタ・ファイルへのアクセスを可能にしたことを特徴とするレジスタファイルアクセス方法。
【請求項５】
前記第１の選択された論理ウィンドウは書込みによりアクセスされるものであり、さらに、前記レジスタファイルアクセス方法は、
前記複数の論理ウィンドウの中から選択し、それにより、任意の時点でのアクセスを第２の選択された論理ウィンドウに制限する過程と、
前記第２の選択された論理ウィンドウのレジスタ群から、所望のレジスタをその仮想レジスタ番号により直接選択する過程と
を含み、この第２の選択された論理ウィンドウが読取りによりアクセスされるものであることを特徴とする請求項４記載の方法。
【請求項６】
少なくとも１つの大域レジスタを設ける過程と、
大域レジスタを任意の時点で必要に応じてアクセスする過程とをさらに含む請求項４記載の方法。
【請求項７】
それぞれ複数のレジスタを備えた複数の論理ウィンドウを備え、少なくとも１つの論理ウィンドウの一部のレジスタが他の論理ウィンドウと共用されているレジスタファイル装置であって、
前記各論理ウィンドウのそれぞれのレジスタを指定するためにそれぞれのレジスタに仮想レジスタ番号が付されているレジスタファイル装置において、
前記複数の論理ウィンドウの中から選択し、それにより、任意の時点でのの書き込みアクセスを前記論理ウィンドウの中の１つの選択された書込みウィンドウに制限する書込みウィンドウ選択手段と、
前記選択された書き込みウィンドウのレジスタ群から、所望のレジスタをその仮想レジスタ番号により直接選択する書き込みレジスタ選択手段と、
前記複数の論理ウィンドウの中から選択し、それにより、任意の時点でのの読取りアクセスを前記論理ウィンドウの中の１つの選択された読取りウィンドウに制限する読取りウィンドウ選択手段と、
前記選択された読取りウィンドウのレジスタ群から、所望のレジスタをその仮想レジスタ番号により直接選択する読取りレジスタ選択手段と、
を設け、仮想レジスタ番号から、実際のレジスタを示す物理レジスタ番号への変換を行

うことなく、レジスタ・ファイルへのアクセスを可能にしたことを特徴とするレジスタファイル装置。

【請求項８】

前記選択された読取りウィンドウ及び選択された書込みウィンドウとは無関係にアクセス可能である少なくとも１つの大域レジスタをさらに具備する請求項７記載の装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】

本発明はコンピュータシステムに関し、特に、マイクロプロセッサにおいてレジスタファイルをアクセスするための論理演算装置に関わる装置及び方法に関する。

【０００２】

【従来の技術】

マイクロプロセッサチップは論理演算装置（ＡＬＵ）と、１つのレジスタファイルとして編成される複数のレジスタとを含む。演算命令、論理命令又はロード／ストア命令などのいくつかの命令によって、ＡＬＵはレジスタファイル中の１つ又は複数のレジスタからデータを読取り、その命令により定義されている演算を実行し、次にその結果をレジスタファイル中の１つ又は複数の指定のレジスタにライトバックする。このシーケンスは実質的には命令ごとに起こるので、システムの総合的なデータ処理速度を考える上でレジスタファイルのアクセス時間は最も重要な問題である。

【０００３】

図１を参照すると、従来の技術によるレジスタアクセス回路のブロック線図が示されている。レジスタアクセスブロック線図３００はＡＬＵ３１０と、レジスタメモリアレイ３３０と、仮想／物理アドレス変換器３２０と、読取りデータ経路３４０と、書込みデータ経路３５０とを含む。読取りデータ経路３４０と書込みデータ経路３５０はＡＬＵ３１０と、レジスタメモリアレイ３３０とにそれぞれ結合している。

【０００４】

従来の技術の１例によれば、レジスタメモリアレイ３３０は本発明の譲受人であるＳｕｎ　Ｍｉｃｒｏｓｙｓｔｅｍｓ，Ｉｎｃ．が開発したＳＰＡＲＣマイクロプロセッサのシリーズの中で使用されているようなウィンドウ形レジスタアレイである。ＳＰＡＲＣシリーズの様々なマイクロプロセッサには、４０個から５２０個までのレジスタがある。全てのＳＰＡＲＣマイクロプロセッサは共通のレジスタファイル編成を共用している。全レジスタのうち８つは大域レジスタとして専用のレジスタである。残るレジスタは各１６個よりなるレジスタセットにグループ分けされている。各セットに含まれる１６個のレジスタのうち８つはＬＯＣＡＬレジスタとして指定され、８つはＯＵＴレジスタとして指定されている。レジスタセットはさらに複数のウィンドウに編成されている。各ウィンドウは８つの大域レジスタと、８つのＩＮレジスタと、８つのＬＯＣＡＬレジスタと、８つのＯＵＴレジスタとを含む。ウィンドウは、１つのウィンドウのＯＵＴレジスタが隣接ウィンドウのＩＮレジスタと物理的に同一のレジスタであるように配列されている。２つの隣接ウィンドウが共用する共通レジスタは異なる仮想アドレスを有するが、実際にはレジスタメモリアレイ中の物理アドレスは共通である。さらに、どのウィンドウも同じ組の大域レジスタを共用する。従って、物理レジスタの数は仮想レジスタの数より少なくなる。この件に関する詳細については、カリフォルニア州メンロパークのＳＰＡＲＣ　Ｉｎｔｅｒｎａｔｉｏｎａｌより入手可能であるＳＰＡＲＣ　Ａｒｃｈｉｔｅｃｔｕｒｅ　Ｍｏｎｕａｌを参照。

【０００５】

レジスタメモリアレイ３３０を特に参照すると、個々のレジスタ$２０_{(1)}$から$２０_{(m)}$はマイクロプロセッサチップにトップダウン方式でＳＲＡＭ配列されるのが典型的である。各レジスタ$２０_{(1)}$から$２０_{(m)}$は（ｎ）ビット幅であり、（ｎ）個のメモリセルを含む。ビット線$２２_{(1)}$～$２２_{(n)}$はメモリアレイの高さに沿って伸びている。各レジスタ$２０_{(1)}$から$２０_{(m)}$からの１つのセルは特定のビット線に結合しており、その特定のビット

線を同様の位置にあるメモリセルと互いに共用し合う。すなわち、たとえば、各レジスタ２０$_{(1)}$から２０$_{(m)}$からの第５のメモリセルはビット線２２$_{(5)}$に結合しているのである。

【０００６】

簡潔を期するため、レジスタメモリアレイ３３０は図１には１つの読取りポートと、１つの書込みポートのみを有するものとして示されている。ところが、典型的には、ＬＡＵ３１０は命令ごとに２つ（以上）のデータ語をレジスタメモリアレイ３３０から読取る。ＡＬＵ３１０がスーパスカラプロセッサである場合、ＡＬＵは一度に２つ以上の命令を実行することが可能であるので、この能力を支援するために、レジスタメモリアレイ３３０は典型的にはいくつかの読取りポートと、いくつかの書込みポートとを有する。さらに、ＡＬＵ３１０がパイプライン化されている場合、すなわち、同時にいくつかの命令の異なる部分について動作する場合には、いずれかの任意の時点で、ＡＬＵ３１０が読取っているレジスタのウィンドウはＡＬＵ３１０が書込んでいるレジスタの同じウィンドウではないということにもなるであろう。従って、パイプライン化プロセッサにおけるいずれかの任意の時点で、レジスタメモリアレイ３３０からの実行パイプラインの開始時に１つの命令に関わる入力を読取るための現在ウィンドウポインタは、メモリアレイ３３０への実行パイプラインの終了時に命令の結果を書込むための現在ウィンドウとは異なる値を有することもありうるだろう。

【０００７】

一つの実施例では、ウィンドウ０からウィンドウ１５とアドレス指定された１６個のウィンドウがある。各々のウィンドウは、３２個のレジスタをアドレス指定できる。レジスタはプログラマにより、現在ウィンドウとは無関係である仮想レジスタ番号を使用してアドレス指定される。この実施例においては、８つの大域レジスタをレジスタ０からレジスタ７としてアドレス指定する。同様に、８つのＯＵＴレジスタをレジスタ８からレジスタ１５としてアドレス指定し、８つのＬＯＣＡＬレジスタをレジスタ１６からレジスタ２３としてアドレス指定し、８つのＩＮレジスタはレジスタ２４からレジスタ３１としてアドレス指定する。この実施例のウィンドウは１６個であり、また、各々のウィンドウは３２個のレジスタをアドレス指定することができるので、メモリアレイ３３０が５１２個のレジスタを有していなければならないことは明白であろう。しかしながら、レジスタは重複使用されるため、メモリアレイ３３０にある物理レジスタの数は５１２個より少ない。この例では、８つの大域レジスタに加えて、ウィンドウ１６個分のレジスタ（ウィンドウごとに８つのＬＯＣＡＬレジスタと、８つのＩＮ／ＯＵＴレジスタがある）があるので、メモリアレイ３３０の中の物理レジスタの総数は実際には２６４である。

【０００８】

仮想／物理変換器３２０は、プログラマが指定した仮想レジスタ番号からメモリアレイ３３０中の物理レジスタ番号への変換を実行する。現在ウィンドウ番号（すなわち、ウィンドウ０から１５）と、そのウィンドウの中における仮想レジスタ番号（すなわち、仮想レジスタ０から３１）とを与えられると、変換器論理３２０は対応する物理レジスタ番号（すなわち、物理レジスタ０から２６４）を与える。尚、１６個のウィンドウがあるので、１つのウィンドウを４ビット数によって指定できる（2^4＝１６）ことに注意する。さらに、１つのウィンドウの中には３２個のレジスタがあるので、ウィンドウ内の特定の１つのレジスタを５ビット仮想レジスタ番号によって指定できる（2^5＝３２）。また、メモリアレイ３３０には２６４個の物理レジスタがあるので、１つの物理レジスタを９ビットの物理レジスタ番号によって指定できる（2^9＝５１２，５１２＞２６４）。

【０００９】

一実施例では、仮想／物理変換器３２０は変換を実行するためにステアリング論理と、加算器とを使用する。この実施例においては、メモリアレイ３３０は初めの８つのレジスタが大域レジスタであるような構造をもつ。そこで、メモリアレイ３３０の次の２５６個のレジスタは特定ウィンドウ所属レジスタとなる。それらの特定ウィンドウ所属レジスタはメモリアレイ３３０の中で昇順ウィンドウ番号によってグループ化されると共に、各ウィ

ンドウの中では昇順仮想レジスタ番号によってさらにグループ化される。この構成では、変換すべき仮想レジスタ番号が大域レジスタの仮想レジスタ番号（すなわち、０から７）であるならば、変換論理３２０が与える物理レジスタ番号は仮想レジスタ番号と同じになる。これに対し、変換すべき仮想レジスタ番号が特定ウィンドウ所属レジスタの仮想レジスタ番号である場合には、変換器３２０が実行する仮想／物理レジスタ変換計算はより複雑になる。まず、変換器３２０は現在ウィンドウポインタを左へ場所４つだけシフトし、次に、シフト後の現在ウィンドウポインタを変換すべき仮想レジスタ番号に加算する。最後に、変換器３２０はその結果の和を取り上げ、モジュロ２５６演算を実行して、変換すべき仮想レジスタ番号に対応する物理レジスタ番号を得る。尚、２５６はウィンドウの数（１６）に各ウィンドウ中の実際の物理レジスタの数（１６）を乗算した数である。ウィンドウの数又は１つのウィンドウの中のレジスタの数が異なる実施例においては、モジュロ演算の基底を相応して調整することになるであろう。

【００１０】

レジスタメモリアレイ３３０から１つのレジスタを読取るために、ＡＬＵ３１０は読取るべきレジスタのウィンドウと、読取るべきレジスタの仮想レジスタ番号とを現在ウィンドウポインタ（読取り）バス３６５及び仮想レジスタ番号（読取り）バス３６０を各々使用して、変換器論理３２０に指定する。次に、変換器論理３２０は読取るべきレジスタのウィンドウと、読取るべきレジスタの仮想レジスタ番号とを物理レジスタ番号に変換し、この物理レジスタ番号を物理レジスタ番号（読取り）バス３７０を使用してメモリアレイ３３０に指定する。読取るべき物理レジスタ番号を受けると、アクセスすべきレジスタの個々のメモリセルのデータ内容が各々のビット線$22_{(1)}$から$22_{(n)}$に置かれ、読取りデータ経路３４０によってＡＬＵ３１０へ転送される。

【００１１】

同様に、メモリアレイ３３０の１つのレジスタに書込むときには、ＡＬＵである機能ユニット３１０は現在ウィンドウポインタ（書込み）バス３８５及び仮想レジスタ番号（書込み）バス３８０を使用して、書込むべきウィンドウと、書込むべきレジスタの仮想レジスタ番号とを変換器論理３２０にそれぞれ指定する。次に、変換器論理３２０は書込むべきウィンドウとレジスタの仮想レジスタ番号を物理レジスタ番号に変換し、このレジスタを物理レジスタ番号（書込み）バス３９０を使用してメモリアレイ３３０に指定する。書込むべきレジスタの物理レジスタ番号を受けると、メモリアレイ３３０は機能ユニット３１０から書込みデータ経路３５０を介してデータの語を受ける。書込むべきこのデータの語をビット線$22_{(1)}$から$22_{(n)}$に置いて、次に、指定のレジスタのメモリセルに記憶させる。

【００１２】

従来の技術のレジスタファイルアクセス構成に関連して、いくつかの問題がある。仮想／物理変換の時間とレジスタメモリアレイ３３０のアクセス時間は非常に長い。これら２つの要因はコンピュータシステムのプロセッサ処理能力を著しくそこなわせる。

【００１３】

レジスタを読取るとき又はレジスタに書込むとき、変換器論理３２０が現在ウィンドウポインタと仮想レジスタ番号を物理レジスタ番号に変換している間に必ず遅延が起こる。典型的な命令は読取るべき２つのレジスタと、書込むべき１つのレジスタとを指定するので、典型的には命令ごとに３回の変換が要求され、各々の変換はそれ独自の遅延を導入する。

【００１４】

レジスタメモリアレイのビット線$22_{(1)}$から$22_{(n)}$は非常に長く、各ビット線に結合するセルの数の関係上、その容量性負荷は大きい。ビット線の容量性負荷が大きいほど、特定のレジスタから読取るのに要する時間又は特定のレジスタに書込むのに要する時間は長くなる。さらに、ビット線の容量性負荷が大きいほど、書込み動作中にレジスタへデータを駆動するために要求されるドライバは大きくなり、また、読取り動作中にレジスタからのデータ出力を感知するために要求されるセンス増幅器は大きくなる。

【００１５】
【発明が解決しようとする課題】
アクセス時間の短い新規なレジスタウィンドウファイル方法とその装置を提供するのが本発明の課題である。
【００１６】
【課題を解決するための手段】
レジスタファイルは複数のレジスタから形成されている。それらのレジスタは複数の論理ウィンドウにグループ化されている。ウィンドウ選択論理は論理ウィンドウの中から選択し、それにより、任意の時点における選択された論理ウィンドウへのアクセスを制限する。
【００１７】
アクセスを一度に１つのウィンドウにのみ限定するので、個々のレジスタの仮想レジスタ番号を指定することにより、そのレジスタを選択できる。従って、レジスタをアクセスするときに仮想レジスタ番号から物理レジスタ番号に変換する必要はない。すなわち、従来の技術の仮想レジスタ番号／物理レジスタ番号変換論理は不要になるのである。従って、以前は変換論理が占めていた集積回路チップ上の領域は必要ではなくなる。さらに、変換論理により命令ごとに導入される変換遅延も排除される。
【００１８】
その上、各レジスタはそのウィンドウのその他のレジスタと読取り線及び書込み線を共用するだけである。従って、レジスタファイルの各ビット線と関連する容量性負荷は、各々のレジスタがレジスタファイルの１つおきのレジスタとビット線を共用していた従来の技術の容量性負荷より著しく小さい。単一のウィンドウのレジスタの中からのみ選択を実行するので、レジスタファイルにデータを書込むとき及びレジスタファイルからデータを読取るときにそれぞれ必要とされるドライバとセンス増幅器はより小型で、それほど強力でないもので良い。
本発明の方法及び装置の目的、特徴及び利点は以下の本発明の詳細な説明から明白になるであろう。
【００１９】
【実施例】
任意の時点で唯一つのウィンドウのレジスタのみを読取り可能とし且つ任意の時点で唯一つのウィンドウのレジスタのみに書込み可能にするという利点をもつウィンドウ形レジスタファイルを実現する方法及び装置を開示する。このようにすると、仮想索引番号を使用してレジスタを直接にアドレス指定することができ、各メモリセルは相対的に短い読取り線と書込み線を有する。従って、レジスタファイルのレジスタに対するアクセスは従来の技術の類似のサイズのレジスタファイルの場合より速い。
【００２０】
以下の説明中、本発明を完全に理解させるために、説明の便宜上、特定の数、材料及び構成を挙げる。しかしながら、それらの特定の詳細がなくとも本発明を実施できることは当業者には明白であろう。別の場合には、本発明を無用にわかりにくくしないために、周知のシステムを概略図又はブロック線図の形態で示す。
【００２１】
図２は、レジスタのアレイから一度に唯一つのウィンドウのレジスタだけをアクセスするためのセレクタの使用を示している。図２では、大域レジスタ４０５と、非大域レジスタ（４２０，４２５，４３０，４３５，４８０，４８５，４９０，４９５など）とは一体となってウィンドウ形レジスタファイルを形成する。簡潔を期するため、図２に示す実施例においては、ウィンドウ形レジスタファイルは１つの読取りポートと、１つの書込みポートのみを有する。
【００２２】
ウィンドウ形レジスタファイルのレジスタは、物理的には、大域レジスタ４０５がまとまって位置し且つ非大域レジスタもまとまって位置しているようにグループ化されている。

さらに、非大域レジスタは物理的にはウィンドウによってグループ化されており、各々のウィンドウの中では、１つのウィンドウのＬＯＣＡＬレジスタがまとまって位置し且つそのウィンドウのＯＵＴレジスタもまとまって位置しているようにグループ化されている。さらに、ウィンドウ０を除いて、ウィンドウのＬＯＣＡＬレジスタは物理的にはそのウィンドウのＯＵＴレジスタと、論理的に先行するウィンドウのＯＵＴレジスタとの間に「はさまって」いる。

【００２３】

先に述べた通り、１つのウィンドウのＩＮレジスタは物理的には論理的に先行するウィンドウのＯＵＴレジスタと同じである。従って、図２の構成では、各ウィンドウのＩＮレジスタと、ＬＯＣＡＬレジスタと、ＯＵＴレジスタとは、ウィンドウ０を除いて、物理的にはまとまってグループ化されている。たとえば、ウィンドウ０のＯＵＴレジスタ４２５はウィンドウ１のＩＮレジスタである。そこで、この図にはＩＮレジスタは示されていない。従って、ウィンドウ１の非大域レジスタはウィンドウ０のＯＵＴレジスタ４２５（すなわち、ウィンドウ１のＩＮレジスタ）と、ウィンドウ１のＬＯＣＡＬレジスタ４３０と、ウィンドウ１のＯＵＴレジスタ４３５とにより形成されることになる。

【００２４】

ウィンドウ０が例外になる原因は、ウィンドウが論理的には円形の構成を成して配列されているが、物理的には矩形アレイとして表現されることである。従って、ウィンドウ０は論理的にはウィンドウ１５に隣接しているのであるが、それらのウィンドウの間には物理的不連続が存在している。そのため、ウィンドウ１５のＯＵＴレジスタ４９５は実際にはウィンドウ０のＩＮレジスタであるが、ウィンドウ１５のＯＵＴレジスタ４９５を物理的にウィンドウ０のＬＯＣＡＬレジスタ４２０と、ウィンドウ１５のＬＯＣＡＬレジスタ４９０の双方に隣接して配置することはできない。

【００２５】

図２は、ウィンドウ１６個分のレジスタがあり且つ各ウィンドウのレジスタは８つの大域レジスタと、８つのＬＯＣＡＬレジスタと、８つのＩＮレジスタと、８つのＯＵＴレジスタとにより形成されているような実施例を提示している。別の実施例においては、ウィンドウの数と、１つのウィンドウの中の各々の型のレジスタの数は図２に提示した数とは異なる。実際、代替実施例の１つでは、大域レジスタは存在せず、別の実施例にはＬＯＣＡＬレジスタがなく、さらに別の実施例にはＩＮ／ＯＵＴレジスタがない。さらに、交代する大域レジスタの組があり、非大域レジスタのウィンドウの中から選択するために使用される方式に類似する方式で特定の１組の大域レジスタを選択するような実施例がある。また、その時点で選択されていない全てのウィンドウのレジスタセットをエネルギー節約用パワーダウンモードに置く実施例もある。しかしながら、それら全ての実施例を統合する概念がある。この概念は、任意の時点でアクセスできるレジスタの数を利用可能なレジスタの総数のうちの１サブセットに制限することにより個々のレジスタに対するアクセスをスピードアップする能力である。

【００２６】

図２のようなレジスタファイルはいくつかのレジスタのウィンドウを含んでいるが、任意の時点で読取られるレジスタのウィンドウは唯一つ（すなわち、現在読取りウィンドウ）であり、任意の時点で書込まれるレジスタのウィンドウも唯一つ（すなわち、現在書込みウィンドウ）である。任意の時点でアクセスできるレジスタの数は少なくなっており、また、任意の時点でアクセスできるレジスタの大部分は物理的にまとまってグループ化されているので、レジスタをアクセスするために使用される線路を短縮することができ、従って、従来の構成で見られたより容量性負荷を少なくすることができる。そのため、本発明のレジスタファイルのレジスタは同じ数のレジスタを有する従来の技術のレジスタファイルのレジスタと比較してより高速でアクセス可能である。

【００２７】

さらに、本発明では、読取り中であるレジスタは現在読取りウィンドウに属し、書込み中であるレジスタは現在書込みウィンドウに属している。従って、現在読取りウィンドウと

現在書込みウィンドウをウィンドウ選択論理に対して暗示することにより、１回のアクセスで指定される物理レジスタ番号をそのアクセスを発生させる命令において指定される仮想レジスタ番号と同一にすることができる。その結果、ウィンドウ及び仮想レジスタの番号を物理レジスタ番号に変換する過程が省略されるので、レジスタへのアクセスは従来の技術と比べて速くなる。

【００２８】

図２においては、ウィンドウ形レジスタファイルに書込むべきデータ語はＤＡＴＡ　ＷＯＲＤ　ＩＮ５１０としてバス５１５を介して書込みレジスタマルチプレクサ（ｍｕｘ）５２５に提示される。ウィンドウレジスタｍｕｘ５２５の各々の入力端子と出力端子は、１語幅である。書込みレジスタｍｕｘ５２５は、１つの入力端子と、１つのウィンドウの中に存在している論理レジスタの数と同じ数の出力端子とを有するマルチプレクサである。制御バスにアサートされた信号－仮想レジスタ番号（書込み）５２０は書込みレジスタｍｕｘ５２５の出力を選択する。各ウィンドウが３２個の論理レジスタを有する実施例では、仮想レジスタ番号（書込み）５２０は書込むべきレジスタの５ビット仮想レジスタ番号を書込みレジスタｍｕｘ５２５に提供する５ビット（2^5＝３２）制御線である。この実施例において、大域レジスタが８つある場合、仮想レジスタ番号（書込み）制御線５２０にアサートされる０から７の値はＤＡＴＡ　ＷＯＲＤ　ＩＮ５１０を出力バス５２９を介して大域レジスタ４０５のうちの適切なレジスタに提示させる。ところが、仮想レジスタ番号（書込み）制御線５２０に８から３１の値がアサートされた場合には、ＤＡＴＡ　ＷＯＲＤ　ＩＮ５１０は２４個の入力端子のうちの適切な１つの入力端子を介して書込みウィンドウｍｕｘ５３５に提示される。

【００２９】

書込みウィンドウｍｕｘ５３５は１つのウィンドウの非大域レジスタごとに１つずつの入力端子を有する。概念の上では、書込みウィンドウｍｕｘ５３５の入力端子ごとに、ウィンドウの数と同じ数の出力端子が存在する。すなわち、ウィンドウが１６個あり且つ各々のウィンドウがいずれも２４個の非大域レジスタを有するような実施例では、書込みウィンドウｍｕｘ５３５は２４個の入力端子と、概念上は３８４個（ウィンドウの数１６×ウィンドウごとの非大域レジスタの数２４＝３８４）の出力端子とを有することになる。しかしながら、実際には、１つのウィンドウのＩＮレジスタの数は物理的には論理的に先行しているウィンドウのＯＵＴレジスタの数と同じであるので、それらの出力端子を組合わせることができる。従って、１つのウィンドウの中に２４個の非大域レジスタがあり且つそれらのレジスタが８つのＩＮレジスタと、８つのＬＯＣＡＬレジスタと、８つのＯＵＴレジスタとに分割されているような実施例においては、書込みウィンドウｍｕｘ５３５は実際には２５６個（ウィンドウの数１６×ウィンドウごとの（ＬＯＣＡＬレジスタの数８＋ＩＮ／ＯＵＴレジスタの数８）＝２５６）の出力端子しかもたない。

【００３０】

制御信号、すなわち現在ウィンドウポインタ（書込み）５３０は、書込みウィンドウｍｕｘ５３５に対する各々の入力が現在書込みウィンドウについて適切な出力端子へチャネリングされるように書込みウィンドウｍｕｘ５３５を制御する。すなわち、１６個のウィンドウがある先の実施例では、現在ウィンドウポインタ（書込み）５３０はそれら１６個のウィンドウの中から選択するために使用される４ビット信号となる。たとえば、現在書込みウィンドウがウィンドウ３である場合、書込みウィンドウｍｕｘ５３５はその入力のうち８つをウィンドウ３の８つのＯＵＴレジスタへチャネリングし、入力のうち８つをウィンドウ３の８つのＬＯＣＡＬレジスタへチャネリングすると共に、入力のうち８つを論理的に先行しているウィンドウ（すなわち、ウィンドウ２）のＯＵＴレジスタへチャネリングする。次に、論理的に先行するウィンドウの８つのＯＵＴレジスタを現在書込みウィンドウの８つのＩＮレジスタとしてアドレス指定する。現在書込みウィンドウがＳＡＶＥ指令の実行によって論理的に次に続くウィンドウ（すなわち、ウィンドウ４）に変更されるか、あるいは、ＲＥＳＴＯＲＥ指令の実行によって論理的に先行するウィンドウ（すなわち、ウィンドウ２）に変更されるまで、書込みウィンドウｍｕｘ５３５はウィンドウ３に

セットされたままである。

【００３１】

読取り時のウィンドウ形レジスタファイルのレジスタからの選択は書込み時の選択に類似しているが、書込み時に対しては鏡像関係となっている。制御信号、すなわち現在ウィンドウポインタ（読取り）５６０は、読取りウィンドウｍｕｘ５６５に対して、制御信号、すなわち現在ウィンドウポインタ（書込み）５３０が書込みウィンドウｍｕｘ５３５に対して示すのと同様の制御機能を有する。読取りウィンドウｍｕｘ５６５は書込みウィンドウｍｕｘ５３５の出力端子５４０と同じ数の入力端子５５０を有する。さらに、読取りウィンドウｍｕｘ５６５は書込みウィンドウｍｕｘ５３５の入力端子５２８と同じ数の出力端子５６７を有する。すなわち、書込みウィンドウｍｕｘ５３５は入力を現在書込みウィンドウに基づいて非大域レジスタ４１０のいくつかのレジスタの中の１つへファンアウトするために使用され、読取りウィンドウｍｕｘ５６５は現在読取りウィンドウに基づいて非大域レジスタ４１０のいくつかのレジスタの中の１つから出力を選択するために使用されるのである。

【００３２】

制御信号、すなわち仮想レジスタ番号（読取り）５７０は読取りレジスタｍｕｘ５７５に対して、制御信号、すなわち仮想レジスタ番号（書込み）５２０が書込みレジスタｍｕｘ５２５に対して示すのと同様の制御機能を有する。読取りレジスタｍｕｘ５７５は書込みレジスタｍｕｘ５２５の出力端子（５２９及び５２８）と同じ数の入力端子（５６７及び５５５）を有する。さらに、読取りレジスタｍｕｘ５７５は単一の出力端子５８０を有し、書込みレジスタｍｕｘ５２５は単一の入力端子５１５を有する。すなわち、書込みレジスタｍｕｘ５２５は単一の入力を仮想レジスタ番号（書込み）５２０に基づいて１つのウィンドウのいくつかのレジスタの中の１つへファンアウトするために使用され、読取りレジスタｍｕｘ５７５は仮想レジスタ番号（読取り）５７０に基づいてウィンドウのいくつかのレジスタの中の１つから単一の出力を選択するために使用されるのである。

【００３３】

図２は、非大域レジスタが物理的にウィンドウによってグループ化されるような実施例を示しているが、異なる方式でレジスタをグループ化することは可能である。別の実施例では、どのウィンドウの類似のレジスタも全て物理的にまとまってグループ化されるように非大域レジスタを配列する。すなわち、たとえば、各ウィンドウの第１のＬＯＣＡＬレジスタの組の後には各ウィンドウの次のＬＯＣＡＬレジスタの組が必ず続き、この配列は各ウィンドウの最後のＯＵＴレジスタの組から構成されるレジスタ群に至るまで続いて行くであろう。このようなレジスタの交互編成によって、どのウィンドウの類似のレジスタの間でも読取り線と書込み線を共用できると共に、共用される線ごとに最小限の長さの線路を使用することができる。

【００３４】

図３は、異なるウィンドウのレジスタによる読取り線と書込み線の共用を示す。図３は、共用読取り線に対して複数の読取りポートが設けられていることと、共用書込み線に対して複数の書込みポートが設けられていることをさらに示している。図３は、図２を大幅に簡略化した構成であり、ウィンドウが３つしかなく各々のウィンドウは単一の１ビットレジスタから構成されている。図３では、バッファ・インバータ対６１０及び６１５は一体となって、ウィンドウ０に関わる１ビットレジスタである１ビットセルを形成している。同様に、バッファ・インバータ対６２０及び６２５はウィンドウ１に関わる１ビットレジスタである１ビットセルを形成し、バッファ・インバータ対６３０及び６３５は一体となって、ウィンドウ２に関わる１ビットレジスタである１ビットセルを形成している。

【００３５】

バッファ及びインバータ６９０はセンス増幅器として動作し、共用読取り線６９５により各々の１ビットセルに結合されている。詳細にいえば、ウィンドウ０に関わる（バッファ及びインバータ６１０及び６１５により形成される）１ビットセルは、トランジスタ６４０により共用読取り線６９５に結合されている。現在読取りウィンドウがウィンドウ０で

あるとき、信号ＲＥＡＤ　Ｗ０＿ＥＮがアサートされて、トランジスタ６４０をターンオンするので、ウィンドウ０に関わる１ビットセルに記憶されているビット値を共用読取り線６９５を介してセンス増幅器６９０に供給することができる。さらに、ウィンドウ１に関わる（バッファ及びインバータ６２０及び６２５により形成されている）１ビットセルは、トランジスタ６５０により共用読取り線６９５に結合されている。現在読取りウィンドウがウィンドウ１であるとき、信号ＲＥＡＤ　Ｗ１＿ＥＮがアサートされ、トランジスタ６５０をターンオンするので、ウィンドウ１に関わる１ビットセルに記憶されているビット値を共用読取り線６９５を介してセンス増幅器６９０に供給することができる。最後に、ウィンドウ２に関わる（バッファ及びインバータ６３０及び６３５により形成されている）１ビットセルは、トランジスタ６６０により共用読取り線６９５に結合されている。現在読取りウィンドウがウィンドウ２であるとき、信号ＲＥＡＤ　Ｗ２＿ＥＮがアサートされて、トランジスタ６６０をターンオンするので、ウィンドウ２に関わる１ビットセルに記憶されているビット値を共用読取り線６９５を介してセンス増幅器６９０に供給することができる。従って、特定の時点で唯一つのウィンドウが読取られるという事実を利用することにより、いくつかのメモリセルの間で単一の読取り線６９５を共用することができるのである。尚、センス増幅器６９０もいくつかのビットセルにより共用され、それにより、メモリセルごとに別個のセンス増幅器を設ける必要をなくしていることに注意する。別の実施例では、ＲＡＭ編成にビットインタリービングを追加することにより、共用ワイヤ長さをさらに短縮できる。

【００３６】

ＡＮＤゲート６４２，６５２及び６６２は一体となって現在読取りウィンドウに関わるデコーダ論理を形成し、所定の時点でトランジスタ６４０，６５０及び６６０のうち多くとも１つのトランジスタがイネーブルされるように保証する。従って、所定の時点で共用読取り線６９５により読取られるのは唯一つのビットセル、すなわち、選択された読取りウィンドウのビットセルのみである。図３においては可能な読取りウィンドウは３つだけであるので、現在読取りウィンドウを２ビット信号（2^2 ＝４）として符号化することができる。信号ＣＲＷ（２：２）と信号ＣＲＷ（１：１）は、それぞれ、現在読取りウィンドウの最上位ビットと、最下位ビットである。そのため、現在読取りウィンドウが０（２進値で００）である場合には、（ＡＮＤゲート６４２から出力される）信号ＲＥＡＤ　Ｗ０＿ＥＮがアサートされ、トランジスタ６４０がターンオンされることがわかる。さらに、現在読取りウィンドウが１（２進値で０１）である場合には、（ＡＮＤゲート６５２から出力される）信号ＲＥＡＤ　Ｗ１＿ＥＮがアサートされ、トランジスタ６５０がターンオンされる。最後に、現在読取りウィンドウが２（２進値で１０）である場合には、（ＡＮＤゲート６６２から出力される）信号ＲＥＡＤ　Ｗ２＿ＥＮがアサートされ、トランジスタ６６０がターンオンされる。

【００３７】

図３において、センス増幅器６９０は２つの読取りポートに結合している。詳細にいえば、センス増幅器６９０はトランジスタ６８０によりＲＥＡＤ　ＰＯＲＴ０に結合され、センス増幅器６９０はトランジスタ６８５によりＲＥＡＤ　ＰＯＲＴ１に結合されている。すなわち、信号ＲＰ０＿ＥＮをアサートして、トランジスタ６８０をターンオンすることにより、ＲＥＡＤ　ＰＯＲＴ０を介してセンス増幅器６９０の出力を読取ることができるのである。他方、信号ＲＰ１＿ＥＮをアサートして、トランジスタ６８５をターンオンすることにより、ＲＥＡＤＰＯＲＴ１を介してセンス増幅器６９０の出力を読取れる。センス増幅器６９０にトランジスタ６８０及び６８５と並列に追加のトランジスタを結合することにより、図３に追加の読取りポートを追加できる。現在ウィンドウレジスタは頻繁には変わらないので、センス増幅器６９０の出力端子でレジスタの値、この場合には単一のビットを利用することが可能であり、その値を読取るための時間はさらに短縮される。

【００３８】

図３において、書込みは、センス増幅器６９０などのセンス増幅器の使用を要求しないという点を除いて、読取りと同様の方式で実行される。詳細にいえば、ウィンドウ０に関わ

る（バッファ及びインバータ６１０及び６１５により形成されている）１ビットセルは、トランジスタ６４５により共用書込み線６０５に結合されている。現在書込みウィンドウがウィンドウ０であるとき、信号ＷＲＩＴＥ　Ｗ０＿ＥＮがアサートされると、トランジスタ６４５がターンオンするので、ウィンドウ０に関わる１ビットセルに共用書込み線６０５を介して供給される１ビット値を記憶させることができる。さらに、ウィンドウ１に関わる（バッファ及びインバータ６２０及び６２５により形成されている）１ビットセルは、トランジスタ６５５により共用書込み線６０５に結合されている。現在書込みウィンドウがウィンドウ１であるとき、信号ＷＲＩＴＥ　Ｗ１＿ＥＮがアサートされると、トランジスタ６５５がターンオンするので、ウィンドウ１に関わる１ビットセルに共用書込み線６０５を介して供給される１ビット値を記憶することが可能になる。また、ウィンドウ２に関わる（バッファ及びインバータ６３０及び６３５により形成されている）１ビットセルはトランジスタ６６５により共用書込み線６０５に結合されている。現在書込みウィンドウがウィンドウ２であるとき、信号ＷＲＩＴＥ　Ｗ２＿ＥＮがアサートされると、トランジスタ６６５をターンオンするので、ウィンドウ２に関わる１ビットセルに共用書込み線６０５を介して供給される１ビット値を記憶することが可能になる。従って、特定の時点で書込まれるウィンドウは唯一つであるという事実を利用することにより、いくつかのメモリセルの間で単一の書込み線６０５を共用できるのである。

【００３９】

ＡＮＤゲート６４７，６５７及び６６７は一体となって現在書込みウィンドウに関わるデコーダ論理を形成し、所定の時点でトランジスタ６４５，６５５及び６６５のうち多くとも１つがイネーブルされるように保証する。従って、所定の時点で共用書込み線６０５を使用して書込まれるのは唯一つのビットセル、すなわち、選択された書込みウィンドウのビットセルだけである。図３において、可能な書込みウィンドウは３つしか存在していないので、現在書込みウィンドウを２ビット信号（2^2＝４）として符号化することができる。信号ＣＷＷ（２：２）と信号ＣＷＷ（１：１）は、それぞれ、現在書込みウィンドウの最上位ビットと、最下位ビットである。そこで、現在書込みウィンドウが０（２進値で００）であるとき、（ＡＮＤゲート６４７から出力される）信号ＷＲＩＴＥ　Ｗ０＿ＥＮがアサートされ、トランジスタ６４５がターンオンされることがわかる。さらに、現在書込みウィンドウが１（２進値で０１）であるときには、（ＡＮＤゲート６５７から出力される）信号ＷＲＩＴＥ　Ｗ１＿ＥＮがアサートされ、トランジスタ６５５がターンオンされる。最後に、現在書込みウィンドウが２（２進値で１０であるときには、（ＡＮＤゲート６６７から出力される）信号ＷＲＩＴＥＷ２＿ＥＮがアサートされ、トランジスタ６６５がターンオンされる。

【００４０】

図３では、共用書込み線６０５は２つの書込みポートに結合している。詳細にいえば、共用書込み線６０５はトランジスタ６７０によりＷＲＩＴＥ　ＰＯＲＴ０に結合されると共に、トランジスタ６７５によりＷＲＩＴＥ　ＰＯＲＴ１に結合されている。すなわち、信号ＷＰ０＿ＥＮをアサートして、トランジスタ６７０をターンオンすることにより、ＷＲＩＴＥ　ＰＯＲＴ０を介する書込みが可能である。他方、信号ＷＰ１＿ＥＮをアサートして、トランジスタ６７５をターンオンすることにより、ＷＲＩＴＥ　ＰＯＲＴ１を介する書込みが可能である。共用書込み線６０５にトランジスタ６７０及び６７５と並列に追加トランジスタを結合することにより、図３に追加の書込みポートを追加できる。

【００４１】

図２と図３を比較すると、図３のビットセルは図２の非大域レジスタに対応していることがわかる。図３のトランジスタ６４０，６５０及び６６０と、ＡＮＤゲート６４２，６５２及び６６２は図２の読取りウィンドウｍｕｘ５６５に相当する。図３のトランジスタ６４５，６５５及び６６５と、ＡＮＤゲート６４７，６５７及び６６７は図２の書込みウィンドウｍｕｘ５３５に相当する。この対応付けは、選択された読取りウィンドウに基づいて、一度に１つのビットセルしか読取りできず且つ選択された書込みウィンドウに基づいて、一度に１つのビットセルしか書込みできないために成立するのである。図３には、１

つのウィンドウの中の特定の１つのレジスタを選択するために使用され、従って、図２の読取りレジスタｍｕｘ５７５又は書込みレジスタｍｕｘ５２５に相当する素子はない。これは、図３がウィンドウごとに単一の１ビットレジスタしかないきわめて簡略化された構成であるためである。

【００４２】

図４は、レジスタファイル中に２つのウィンドウがあり、各々のウィンドウは２つのレジスタを有し且つ各レジスタは２つのビットを記憶する本発明のさらに複雑な実施例を示す。ウィンドウの選択及び選択されたウィンドウの中におけるレジスタの選択の概念をきわだたせるために、図４からは不要な詳細を省略してある。従って、図４においては、バッファ・インバータ対として１ビットセルを示すのではなく、各々の１ビットセルはブロック線図の形（すなわち、ブロック７１０，７２０，７３０，７４０，７５０，７６０，７７０及び７８０として）示されている。さらに、読取りと書込みは対称であるため、書込み選択論理のみを示した。復号と多重ポート選択の論理も図４には示されていない。

【００４３】

図４では、各レジスタは２ビットレジスタである。従って、各レジスタは２つの１ビット記憶セルにより形成されており、一方の１ビット記憶セルはレジスタの最下位ビット（すなわち、ビット０）を記憶し、他方の１ビット記憶セルは最上位ビット（すなわち、ビット１）を記憶する。すなわち、ブロック７１０はウィンドウ０に関わるレジスタ０のビット０を記憶する１ビット記憶セルであり、ブロック７３０は対応するビット１を記憶する１ビット記憶セルである。ブロック７２０とブロック７４０は一体となってウィンドウ１のレジスタ０を形成し、ブロック７２０はビット０を記憶し、ブロック７４０はビット１を記憶している。同様に、ブロック７５０はウィンドウ０に関わるレジスタ１のビット０を記憶する１ビット記憶セルであり、また、ブロック７７０は対応するビット１を記憶する１ビット記憶セルである。ブロック７６０とブロック７８０は一体となってウィンドウ１のレジスタ１を形成し、ブロック７６０はビット０を記憶し、ブロック７８０はビット１を記憶している。

【００４４】

図２と図４を比較すると、図４のビットセル（すなわち、ブロック７１０，７２０，７３０，７４０，７５０，７６０，７７０及び７８０）は図２の非大域レジスタ４１０に相当することがわかる。図４のトランジスタ７１５，７２５，７３５，７４５，７６５，７７５及び７８５は図２の書込みウィンドウｍｕｘ５３５に相当する。図４のトランジスタ７９０，７９２，７９４及び７９８は図２の書込みレジスタｍｕｘ５２５に相当する。従って、選択された書込みウィンドウと、選択された書込みウィンドウの中の選択されたレジスタとに基づいて、一度に１つのレジスタを書込むことしかできない。このことは例によって最も良く示される。

【００４５】

現在書込みウィンドウがウィンドウ０である場合、信号ＷＩＮＤＯＷ０＿ＥＮはアサートされるであろうが、信号ＷＩＮＤＯＷ１＿ＥＮはアサートされないであろう。そのため、トランジスタ７１５，７５５，７３５及び７７５はターンオンするであろうが、トランジスタ７２５，７６５，７４５及び７８５はターンオフするであろう。ウィンドウ０のレジスタ０に書込むべきであれば、信号ＲＥＧＩＳＴＥＲ０＿ＥＮはアサートされるが、信号ＲＥＧＩＳＴＥＲ１＿ＥＮはアサートされないであろう。その結果、トランジスタ７９０及び７９２はターンオンし、トランジスタ７９４及び７９６はターンオフするであろう。従って、共用書込み線７９１のＢＩＴ０値はブロック７１０に記憶され、共用書込み線７９３のＢＩＴ１値はブロック７３０に記憶されるであろう。そこで、ウィンドウ０のレジスタ１に書込むことが望まれると、信号ＲＥＧＩＳＴＥＲ１＿ＥＮはアサートされるが、信号ＲＥＧＩＳＴＥＲ０＿ＥＮはアサートされないであろう。この結果、トランジスタ７９４及び７９６はターンオンし、トランジスタ７９０及び７９２はターンオフするであろう。従って、共用書込み線７９１のＢＩＴ０値はブロック７５０に記憶され、共用書込み線７９３のＢＩＴ１値はブロック７７０に記憶されるであろう。

【００４６】
現在書込みウィンドウがウィンドウ１に変わった場合、信号ＷＩＮＤＯＷ１＿ＥＮはアサートされるが、信号ＷＩＮＤＯＷ０＿ＥＮはアサートされないであろう。従って、トランジスタ７２５，７６５，７４５及び７８５はターンオンし、トランジスタ７１５，７５５，７３５及び７７５はターンオフするであろう。ウィンドウ１のレジスタ０に書込むべきであるならば、信号ＲＥＧＩＳＴＥＲ０＿ＥＮはアサートされるが、信号ＲＥＧＩＳＴＥＲ１＿ＥＮはアサートされないであろう。この結果、トランジスタ７９０及び７９２はターンオンし、トランジスタ７９４及び７９６はターンオフするであろう。従って、共用書込み線７９１のＢＩＴ０値はブロック７２０に記憶され、共用書込み線７９３のＢＩＴ１値はブロック７４０に記憶されるであろう。そこで、ウィンドウ１のレジスタ１への書込みが望まれたならば、信号ＲＥＧＩＳＴＥＲ１＿ＥＮはアサートされるが、信号ＲＥＧＩＳＴＥＲ０＿ＥＮはアサートされないであろう。この結果、トランジスタ７９４及び７９６はターンオンし、トランジスタ７９０及び７９２はターンオフするであろう。従って、共用書込み線７９１のＢＩＴ０値はブロック７６０に記憶され、共用書込み線７９３のＢＩＴ１値はブロック７８０に記憶されるであろう。

【００４７】
図４では、各々のレジスタは単一のウィンドウでのみアクセス可能であるので、レジスタはＬＯＣＡＬレジスタであるかのように機能する。図５は、１つのウィンドウのＩＮレジスタを論理的に先行するウィンドウのＯＵＴレジスタとしてアドレス指定する技法を示す。この図においても、ある１つの時点で活動しているのは１つのウィンドウのレジスタのみであるという意味で、１つのレジスタを１つのウィンドウのＩＮレジスタとして処理すると共に、別のウィンドウのＯＵＴレジスタとして処理する概念をきわ立たせるために、不要な詳細を省略してある。従って、図５では、１ビットセルをバッファ・インバータ対として示すのではなく、各々の１ビットセルをブロック線図の形で（すなわち、ブロック８１０，８２０及び８３０として）示している。さらに、読取りと書込みは対称であるため、書込み選択論理のみを示す。復号と多重ポート選択の論理も図５には示されていない。

【００４８】
図５の実施例では、ウィンドウは３つ（すなわち、ウィンドウ０，ウィンドウ１及びウィンドウ２）あり、各々のウィンドウは２つのレジスタ（すなわち、ＩＮレジスタ及びＯＵＴレジスタ）を有し、各レジスタは１つのビットのみを記憶する。１つのウィンドウのＩＮレジスタは論理上先行しているウィンドウのＯＵＴレジスタである。図５の実施例のような３つのウィンドウを含む例においては、ウィンドウ２は論理の上でウィンドウ０に先行し、ウィンドウ０は論理の上でウィンドウ１に先行し、ウィンドウ１は論理の上でウィンドウ２に先行している。従って、図５では、ブロック８１０は現在書込みウィンドウがウィンドウ０であるときはウィンドウ０のＩＮレジスタとして扱われ、現在書込みウィンドウがウィンドウ２であるときにはウィンドウ２のＯＵＴレジスタとして扱われるのである。さらに、ブロック８２０は現在書込みウィンドウがウィンドウ１であるときはウィンドウ１のＩＮレジスタとして扱われ、現在書込みウィンドウがウィンドウ０であるときにはウィンドウ０のＯＵＴレジスタとして扱われる。最後に、ブロック８３０は現在書込みウィンドウがウィンドウ２であるときはウィンドウ２のＩＮレジスタとして扱われ、現在書込みウィンドウがウィンドウ１であるときにはウィンドウ１のＯＵＴレジスタとして扱われる。

【００４９】
図５の実施例では、現在書込みウィンドウがウィンドウ０であれば、信号ＷＩＮＤＯＷ０＿ＥＮはアサートされ、信号ＷＩＮＤＯＷ１＿ＥＮ及びＷＩＮＤＯＷ２＿ＥＮはアサートされない。従って、現在書込みウィンドウがウィンドウ０であれば、トランジスタ８１３及び８２７（ＷＩＮＤＯＷ０＿ＥＮにより制御される）はターンオンし、トランジスタ８２３及び８３７（ＷＩＮＤＯＷ１＿ＥＮにより制御される）と、トランジスタ８３３及び８１７（ＷＩＮＤＯＷ２＿ＥＮにより制御される）とはターンオフする。したがって現在

書込みウィンドウがウィンドウ０であるとき、信号ＩＮ＿ＥＮをアサートし且つ信号ＯＵＴ＿ＥＮをアサートしないと、ウィンドウ０のＩＮレジスタをアクセスすることになる。すなわち、トランジスタ８５０（ＩＮ＿ＥＮにより制御される）はターンオンし、トランジスタ８４０（ＯＵＴ＿ＥＮにより制御される）はターンオフするので、ＢＩＴ０がどのような値を有していても、その値はブロック８１０に記憶される。これに対し、現在書込みウィンドウがウィンドウ０であるときに、信号ＯＵＴ＿ＥＮをアサートし且つ信号ＩＮ＿ＥＮをアサートしないと、ウィンドウ０のＯＵＴレジスタをアクセスすることになる。すなわち、トランジスタ８４０（ＯＵＴ＿ＥＮにより制御される）はターンオンし、トランジスタ８５０（ＩＮ＿ＥＮにより制御される）はターンオフするので、ＢＩＴ０がどのような値を有していても、その値はブロック８２０に記憶される。

【００５０】

さらに、現在書込みウィンドウがウィンドウ１であるときには、信号ＷＩＮＤＯＷ１＿ＥＮはアサートされるが、信号ＷＩＮＤＯＷ２＿ＥＮ及びＷＩＮＤＯＷ０＿ＥＮはアサートされない。従って、現在書込みウィンドウがウィンドウ１であるときには、トランジスタ８２３及び８３７（ＷＩＮＤＯＷ１＿ＥＮにより制御される）はターンオンし、トランジスタ８３３及び８１７（ＷＩＮＤＯＷ２＿ＥＮにより制御される）と、トランジスタ８１３及び８２７（ＷＩＮＤＯＷ０＿ＥＮにより制御される）とはターンオフする。現在書込みウィンドウがウィンドウ１であるときに、信号ＩＮ＿ＥＮをアサートし且つ信号ＯＵＴ＿ＥＮをアサートしないと、ウィンドウ１のＩＮレジスタをアクセスする。トランジスタ８５０（ＩＮ＿ＥＮにより制御される）はターンオンし、トランジスタ８４０（ＯＵＴ＿ＥＮにより制御される）はターンオフする。従って、ＢＩＴ０がどのような値を有していても、その値はブロック８２０に記憶される。これに対し、現在書込みウィンドウがウィンドウ１であるときに、信号ＯＵＴ＿ＥＮをアサートし且つ信号ＩＮ＿ＥＮをアサートしないと、ウィンドウ１のＯＵＴレジスタをアクセスする。すなわち、トランジスタ８４０（ＯＵＴ＿ＥＮにより制御される）はターンオンし、トランジスタ８５０（ＩＮ＿ＥＮにより制御される）はターンオフして、ＢＩＴ０がどのような値を有していても、その値はブロック８３０に記憶される。

【００５１】

最後に、現在書込みウィンドウがウィンドウ２であるときには、信号ＷＩＮＤＯＷ２＿ＥＮはアサートされるが、信号ＷＩＮＤＯＷ０＿ＥＮ及びＷＩＮＤＯＷ１＿ＥＮはアサートされない。従って、現在書込みウィンドウがウィンドウ２であるとき、トランジスタ８３３及び８１７（ＷＩＮＤＯＷ２＿ＥＮにより制御される）はターンオンし、トランジスタ８１３及び８２７（ＷＩＮＤＯＷ０＿ＥＮにより制御される）と、トランジスタ８２３及び８３７（ＷＩＮＤＯＷ１＿ＥＮにより制御される）とはターンオフする。現在書込みウィンドウがウィンドウ２であるとき、信号ＩＮ＿ＥＮをアサートし且つ信号ＯＵＴ＿ＥＮをアサートしないと、ウィンドウ２のＩＮレジスタをアクセスする。すなわち、トランジスタ８５０（ＩＮ＿ＥＮにより制御される）はターンオンし、トランジスタ（ＯＵＴ＿ＥＮ）はターンオフして、ＢＩＴ０がどのような値を有していても、その値はブロック８３０に記憶される。これに対し、現在書込みウィンドウがウィンドウ２であるとき、信号ＯＵＴ＿ＥＮをアサートし且つ信号ＩＮ＿ＥＮをアサートしないと、ウィンドウ２のＯＵＴレジスタをアクセスする。すなわち、トランジスタ８４０（ＯＵＴ＿ＥＮにより制御される）はターンオンし、トランジスタ８５０（ＩＮ＿ＥＮにより制御される）はターンオフしてＢＩＴ０がどのような値を有していても、その値はブロック８１０に記憶される。

【００５２】

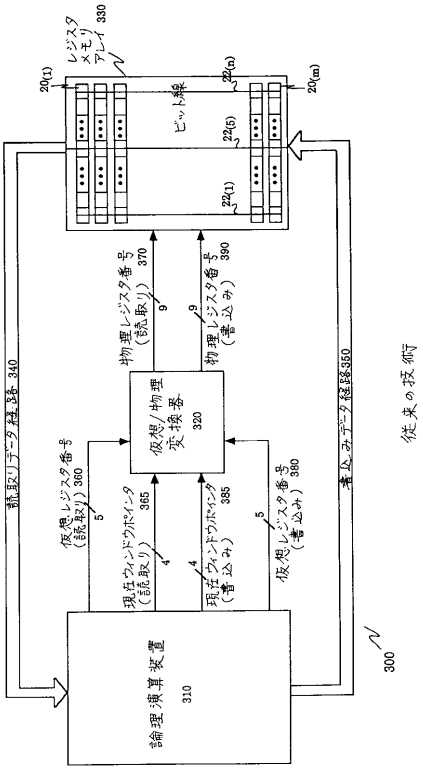
本発明の方法及び装置をその現時点で好ましい実施例及び代替実施例によって説明したが、本発明を特許請求の範囲の趣旨の範囲内で変形及び変更を伴って実施しうることは当業者には認められるであろう。従って、明細書及び図面は限定的な意味をもつのではなく、例示としてみなされるべきである。

【図面の簡単な説明】

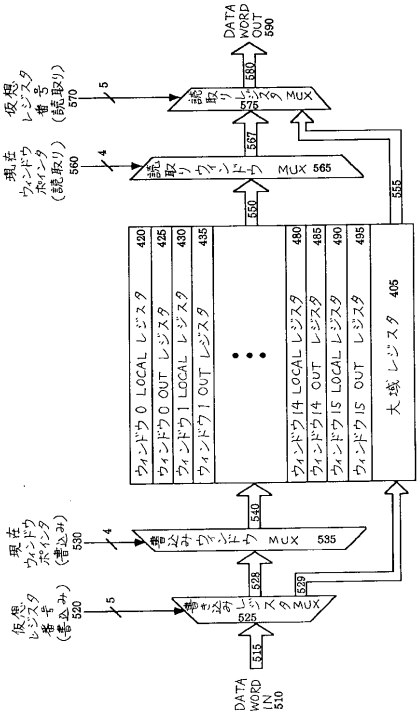
【図１】　ＳＰＡＲＣレジスタウィンドウを使用してレジスタから成るメモリアレイから

データを読取り且つメモリアレイにデータを書込む論理演算装置のブロック線図。
【図２】　レジスタのアレイから一度に唯一つのレジスタのウィンドウをアクセスするためのセレクタの使用を示す図。
【図３】　複数の読取りポートを有する読取り線の異なるウィンドウのレジスタによる共用と、複数の書込みポートを有する書込み線の異なるウィンドウのレジスタによる共用とを示す図。
【図４】　レジスタの現在ウィンドウの選択と、現在ウィンドウの中における異なるウィンドウの選択とを示す図。
【図５】　１つのウィンドウのＩＮレジスタの、論理上先行しているウィンドウのＯＵＴレジスタとしてのアクセスを示す図。
【符号の説明】
４０５…大域レジスタ、４１０，４２０，４２５，４３０，４３５，４８０，４８５，４９０，４９５…非大域レジスタ、５１５…バス、５２０…仮想レジスタ番号（書込み）、５２５…書込みレジスタｍｕｘ、５３０…現在ウィンドウポインタ（書込み）、５３５…書込みウィンドウｍｕｘ、５６０…現在ウィンドウポインタ（読取り）、５６５…読取りウィンドウｍｕｘ、５７０…仮想レジスタ番号（読取り）、５７５…読取りレジスタｍｕｘ。

【図１】



【図２】



【図３】

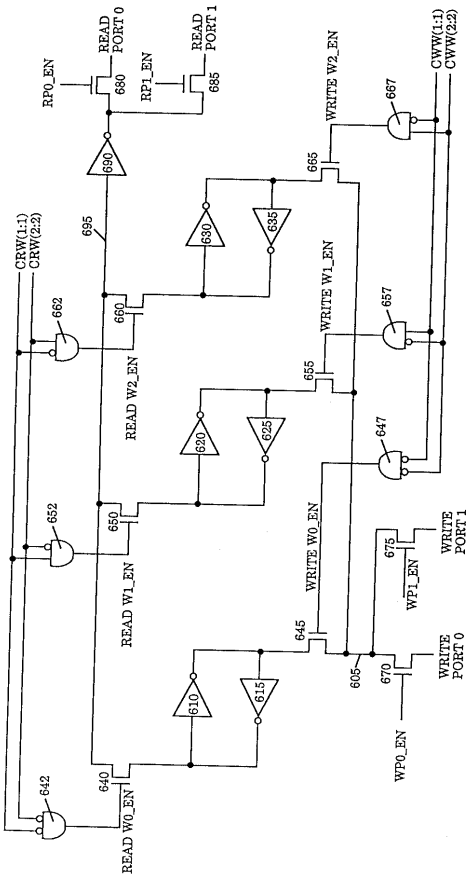


Figure 3

【図４】

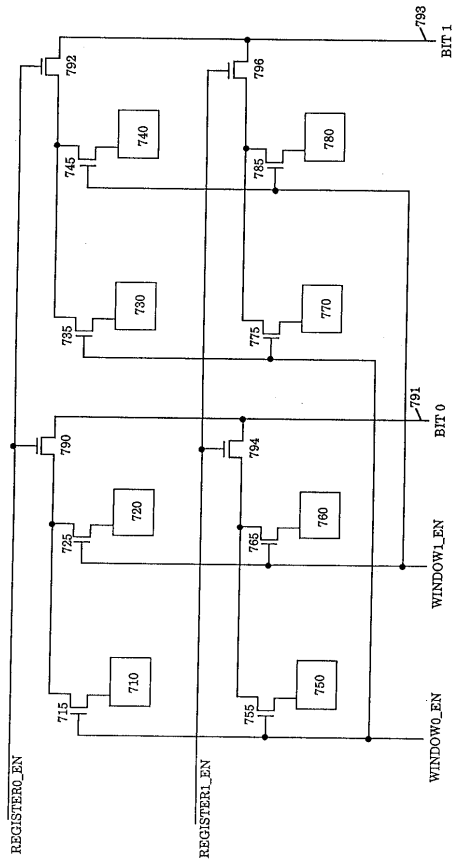


Figure 4

【図５】

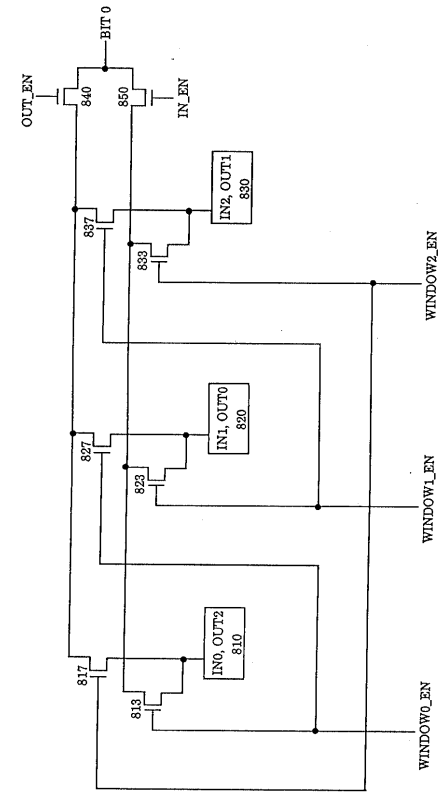


Figure 5

フロントページの続き

(72)発明者　ウィリアム・エヌ・ジョイ
アメリカ合衆国　８１６１２　コロラド州・アスペン・ピイオーボックス２３・（番地なし）
(72)発明者　マイケル・アレン
アメリカ合衆国　９４３０１　カリフォルニア州・サンフランシスコ・ページ　ストリート・８５０
(72)発明者　マーク・トレンブレイ
アメリカ合衆国　９４３０１　カリフォルニア州・パロ　アルト・ウェーバリー　ストリート　ナンバー３・８１０

審査官　後藤　彰

(56)参考文献　特開昭６３－２４５５２９（ＪＰ，Ａ）
特開平４－２４５３２４（ＪＰ，Ａ）
特開平５－３４１９９１（ＪＰ，Ａ）
特開平７－９８６５５（ＪＰ，Ａ）
特開平５－２８２１４７（ＪＰ，Ａ）
特開平５－２００１０（ＪＰ，Ａ）
特開平３－３５３２４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 9/42
G06F 9/34